"SNEC" – The SuperNova Explosion Code

Viktoriya Morozova[1], Christian D. Ott[1], and Anthony L. Piro[2]

[1] TAPIR, Walter Burke Institute for Theoretical Physics, Caltech, Pasadena, CA, USA
[2] Carnegie Observatories, Pasadena, CA, USA

This document is dated April 15, 2015. Check back frequently for updated versions!

Abstract

We present the SuperNova Explosion Code (SNEC), an open-source Lagrangian code that solves for the hydrodynamics and equilibrium-diffusion radiation transport in the expanding envelopes of core-collapse supernovae (CCSNe), taking into account recombination effects and the presence of radioactive nickel. Given a model of the progenitor star and an explosion energy, the code generates the bolometric light curve, as well as the light curves in different observed wavelength bands in the blackbody approximation.

Contents

1 Basic Equations

SNEC solves the equations of Langrangian hydrodynamics in spherical symmetry (using artificial visosity, e.g., [1, 2]), supplement with radiative diffusion (e.g. [3, 4]).

SNEC implements the mass conservation equation,

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho} \,, \tag{1}$$

the energy conservation equation,

$$\frac{\partial \epsilon}{\partial t} = \frac{P}{\rho}\frac{\partial \ln \rho}{\partial t} - 4\pi r^2 Q \frac{\partial v}{\partial m} - \frac{\partial L}{\partial m} + \epsilon_{\mathrm{Ni}} \,, \tag{2}$$

and the momentum conservation equation,

$$\frac{\partial v}{\partial t} = -\frac{Gm}{r^2} - 4\pi r^2 \frac{\partial P}{\partial m} - 4\pi \frac{\partial (r^2 Q)}{\partial m} \,. \tag{3}$$

Here $m = \int_0^r 4\pi r'^2 \rho(r') dr'$ is the mass coordinate, r is the radius, t is the time, ρ is the mass density, ϵ is the specific internal energy (energy per unit mass), P is the pressure, $v = \partial r / \partial t$ is the velocity of the matter, Q is the artificial viscosity, ϵ_{Ni} is the specific energy deposited due to the radioactive decay of ^{56}Ni (see §4), and G is the gravitational constant. The radiative luminosity L is defined as

$$L = -(4\pi r^2)^2 \frac{\lambda a c}{3\kappa}\frac{\partial T^4}{\partial m} \,, \tag{4}$$

where a is the radiation constant, c is the speed of light, λ is the flux-limiter and κ is the Rosseland mean opacity (see §3.3). The expression for flux-limiter is chosen based on the prescription of [5] (see also [6]) as[1]

$$\lambda = \frac{6 + 3R}{6 + 3R + R^2} \,, \tag{5}$$

where

$$R = \frac{4\pi r^2}{\kappa T^4} \left| \frac{\partial T^4}{\partial m} \right| \,. \tag{6}$$

The inclusion of the radiation diffusion term ($\partial L / \partial m$ in the energy equation) is optional, i.e., SNEC can be run either in pure hydro or in radiation-hydro mode. The pure hydro solver is implemented in `hydro.F90`, while `hydro_rad.F90` contains the combined diffusion and hydro solver.

2 Basic Control and Initial Setup

2.1 The `parameters` file

Much of what SNEC does can be controlled via settings read in by `input_parser.F90` from a text file called `parameters` in the code's main directory (most of the parameters have self-explanatory

[1] Note, that our Eq. 5 is 3 times larger than the original expression of [5], and this is compensated by the factor 3 in the denominator of Eq. 4.

names), but using grep with the parameter of interest as an argument on the Fortran files in the src directory can be useful at times to understand what a given parameter really does.

The parameter outdir sets the name of the output directory, which must be created by the user in the main directory before running the code (if the output directory is absent, the code will return a corresponding error message). *The user needs to remember that with every new run, SNEC deletes the previous content of this directory.* The parameter profile_name sets the name of the progenitor profile file that is read in by read_profile.F90 and mapped to SNEC's grid. SNEC uses the .short format whose details can be inferred from read_profile.F90. This is the same format used by the open-source code GR1D [7, 8]. This profile just contains the basic hydrodynamics and thermodynamics of the star. The parameter comp_profile_name sets the name of the profile file containing detailed compositional information for the progenitor. These data are read and mapped by read_profile_compositions.F90 from which the detailed content of the isotope profile file can be inferred (see also §2.2.2 for more details on how the profile data are read in and mapped). We provide a number of example progenitor profiles in the profile directory of the main SNEC code directory. There is also a README.txt file describing them in some more detail.

The parameters responsible for the type of explosion and its setup are described in §2.2.4. The parameter imax sets the number of grid cells to be used in the simulation (see §2.2.1 for a description of the grid setup). The parameter radiation, if set to 1, turns on radiative diffusion (2). The equation of state (EOS) to be used is specified by parameter eoskey (see §5). Parameter Ni_switch activates heating by radioactive nickel (see §2.2.2 and §4). Recombination is traced for the first saha_ncomps isotopes of the composition profile, while saha_ncomps=0 implies full ionization. The mechanism of boxcar_smoothing is discussed in §2.2.3. The setup of the opacity floor (see §3.3) is controlled by parameters opacity_floor_envelope and opacity_floor_core.

2.2 Setup of the Explosion

subroutine problem (in problem.F90) is one of the central routines of the code. It controls the initial setup of the explosion based on the parameters provided by user. It controls the allocation of memory, initialization of variables, reading of the progenitor star profile, grid setup, and set up of the explosion type.

2.2.1 Grid Setup

SNEC's numerical grid is divided into imax computational cells. Quantities with subscript i live at the inner edge of computational cell i (such as velocity v, luminosity L, flux-limiter λ). Quantities with subscript $i + 1/2$ live at the center (in mass coordinate) of cell i (such as specific internal energy ϵ, pressure P, density ρ, temperature T, opacity κ, and artificial viscosity Q). The physical surface of the simulation domain is located at the inner edge of the imax-th cell.

SNEC currently has two options for the gridding:

- uniform gridding in mass,
- customized gridding in mass based on a file provided by the user (option "from_file_by_mass" for gridding in the parameters file).

In order to use the second option, the user must provide a file with name GridPattern.dat in the main directory of the code, containing a column of imax consecutive real numbers in the range

$[0, 1]$. The subroutine `grid` in file `grid.F90` will use it as a pattern for distributing the total mass among the computational cells. Typically, one would like to have grid setup with better resolution of the inner regions (where the bomb is placed) and the surface region of the stellar model (where the photosphere will be initially).

- The mass coordinate m_i is the mass enclosed by the inner boundary of cell i. In the code it is given by the array `mass`.
- For the simulations of core-collapse supernova light curves, one generally needs to have the inner boundary not at the $m_1 = 0$, but at some mass $M > 0$ to account for the collapse of the core to a protoneutron star. The parameter `mass_excised` gives the mass coordinate of the inner boundary (m_1) in solar masses (typically we use $1.4\,M_\odot$), provided `mass_excision = 1`. Otherwise, the inner boundary of the grid coincides with the inner boundary of the progenitor star model in `profile_name`.
- The mass contained in each cell is

  $$\Delta m_{i+1/2} = m_{i+1} - m_i \ , \tag{7}$$

  for $i = 1, 2, ..., \text{imax} - 1$. $\Delta m_{i+1/2}$ is stored in array `delta_mass`. At the outer boundary, we set $\Delta m_{\text{imax}+1/2} = \Delta m_{\text{imax}-1/2}$.
- The mass enclosed by the center of each cell is equal to

  $$m_{i+1/2} = m_i + \frac{1}{2}\Delta m_{i+1/2} \ , \tag{8}$$

  for $1, 2, ..., \text{imax}$. $m_{i+1/2}$ is stored in array `cmass`.
- The mass difference between two consecutive cell centers is given by

  $$\Delta m_i = m_{i+1/2} - m_{i-1/2} = \frac{1}{2}(\Delta m_{i-1/2} + \Delta m_{i+1/2}) \ , \tag{9}$$

  for $i = 2, 3, ..., \text{imax}$. Δm_i is stored in array `delta_cmass`.

2.2.2 Mapping in Progenitor Star Data

In order to explode a star and generate a light curve, SNEC needs information about the structure and chemical composition of the preexplosion star. These data are read in from two separate files, specified as strings in the parameter file under `profile_name` and `comp_profile_name`.

The file specified in `profile_name` has the common `.short` format (e.g., used also by the open-source code `GR1D` [7, 8]). It's format is:

- First line: number of cells/rows in the profile (integer).
- Subsequent lines:

```
cell index   mass   radius   temperature   density   velocity   Ye   Omega
```

All units are in CGS, `zone index` is an integer, the rest of the variables are real numbers and self-explanatory.

The file whose file name must be set in `comp_profile_name` should contain compositional information. Its format is specific to SNEC and described in the following:

- First line: `ncells   nisotopes`
  where `ncells` is the number of cells/rows in the profile file and `nisotopes` is the number of isotopes present in the compositional data.
- Second line: The mass numbers of all isotopes (space-separated real numbers).
- Third line: The charge numbers of all isotopes (space-separated real numbers)
- Subsequent lines:

```
cell mass   cell radius   (Mass fraction of isotope i) for i=1,nisotopes
```

The subroutine `read_profile` (in source file `read_profile.F90`) and subroutine `read_profile_compositions` (in source file `read_profile_compositions.F90`) read in the progenitor profile data and map them on the computational grid using linear interpolation. Mass coordinates are mapped to the inner edges of the computational cells, while temperature, density, velocity and mass fractions are mapped to the cell centers. The specific internal energy and the pressure are found at cell centers by calling the EOS with density, temperature, and compositional information. The outer boundary data (at the center of the last computational cell) are set as described in §3.4. For consistency, SNEC recalculates the radial coordinates from mass coordinates and cell densities in subroutine `integrate_radius_initial` (in `read_profile.F90`).

The present version of SNEC does not include a nuclear reaction network, which would allow the explosion to self-consistently generate radioactive ^{56}Ni (albeit in spherically-symmetric setting). ^{56}Ni is the crucial ingredient that powers the lightcurves of Type I SNe and the late-lightcurves of Type II SNe.

SNEC allows the user to introduce a desired amount of ^{56}Ni by hand. For this to work, the composition profile must have a column for the ^{56}Ni isotope ($Z = 28$, $A = 56$), even if that column is empty in the original progenitor star model. SNEC then uniformly distributes the amount of ^{56}Ni given by the parameter `Ni_mass` between the inner boundary (m_1) and a chosen outer `Ni_boundary_mass` (in the `parameters` file). After this procedure, the mass fractions are renormalized to one at fixed ^{56}Ni mass fraction. The optional boxcar averaging (see next section §2.2.3) smoothens the often sharp compositional gradients of the progenitor profile to emulate mixing during the explosion.

2.2.3 Initial Smoothing of the Compositional Profile

The subroutine `boxcar` is called from the `read_profile_compositions` subroutine. It smoothes the composition profile of the progenitor star. The actual mixing of the chemical isotopes in supernovae is expected to be a result of Rayleigh-Taylor and Richtmyer-Meshkov instabilities, occuring during the propagation of the explosion shock wave through the star. These instabilities can be accurately modeled only in high-resolution three-dimensional simulations that fare beyond SNEC's present scope. In order to mimic the mixing in the eject in spherically-symmetric explosions modeled with SNEC, we use the 'boxcar' averaging.

The nominal width (in mass coordinate) of the boxcar is defined by the hardcoded parameter `boxcar_nominal_mass` in `boxcar.F90` ($0.4\,M_\odot$ is the current default). Starting from the first grid point, the boxcar is moved outwards cell-by-cell, and averaging is performed over all cells contained within the boxcar. Since (in most scenarios) the discretization will be non-uniform in mass, one cannot expect the nominal mass of the boxcar to be precisely contained within an integer number of cells. Due to this, at each shift of the boxcar outwards from a cell i (at the boxcar's inner), the algorithm finds the minimum number of subsequent $(j > i)$ cells that contains the nominal mass, and then averages between them. The actual width of the boxcar is therefore equal to

$$\Delta M = \sum_{l_{\min}}^{l_{\max}} \Delta m_l \, , \tag{10}$$

where $l = l_{\min}, ..., l_{\max}$ are the indices of the mass bins Δm_l inside the boxcar (SNEC stores ΔM in variable `boxcar_actual_mass`; `boxcar_actual_mass` $\geq$ `boxcar_nominal_mass`). The total mass of the k-th isotope in the boxcar is equal to

$$\Delta M_k = \sum_{l_{\min}}^{l_{\max}} X^{\mathrm{old}}_{l,k} \Delta m_l \, , \tag{11}$$

where $X^{\mathrm{old}}_{l,k}$ are the original mass fractions of the element in cells $l = l_{\min}, ..., l_{\max}$.

The new, averaged mass fractions within the boxcar are

$$X^{\mathrm{new}}_{l,k} = \frac{\Delta M_k}{\Delta M} \, , \qquad l = l_{\min}, ..., l_{\max} \, . \tag{12}$$

Mixing in this way conserves the mass in a given isotope within the boxcar:

$$\sum_{l_{\min}}^{l_{\max}} X^{\mathrm{new}}_{l,k} \Delta m_l = \frac{\Delta M_k}{\Delta M} \sum_{l_{\min}}^{l_{\max}} \Delta m_l = \Delta M_k \, , \tag{13}$$

and ensures that the mass fractions sum to unity in every cell,

$$\sum_k X^{\mathrm{new}}_{l,k} = \left(\sum_k \Delta M_k \right) / \Delta M = 1 \, . \tag{14}$$

SNEC repeats the averaging procedure until a desired degree of smoothness is reached. We currently use four sweeps of averaging in SNEC. The number of boxcar sweeps is hardcoded in `boxcar.F90` in parameter `number_iterations`.

2.2.4 Explosion Type

SNEC is set up for thermal bomb (`initial_data = "Thermal_Bomb"`) and piston-driven (`initial_data = "Piston_Explosion"`) explosions.

In the case of a **piston-driven explosion**, the first two computational cells of the profile are boosted outwards with velocity piston_vel (in $\mathrm{cm\,s^{-1}}$) for a time interval deliminated by parameters (set in file parameters) piston_tstart and piston_tend (both in units of seconds). In the piston-driven case it is not straightforward to compute a priori the exact energy given to the exploding star. If one aims for a specific explosion energy, currently the only way to arrive there is to do this iteratively by computing an explosion to free expansion and reading of the final explosion energy from the output provided in file conservation.dat in the output directory.

The format of conservation.dat in the present version of SNEC is as follows:

```
time    Egrav    Eint    Ekin    Etot    EtotmInt
```

where Egrav is the (negative) gravitational energy, Eint is the internal energy, Ekin is the kinetic energy, Etot = Egrav + Eint + Ekin and EtotmInt = Etot - Eint. SNEC presently does not account for energy lost to photons from the photosphere.

In the case of a thermal-bomb driven explosion, the injected total bomb energy E_{tot} (stored by SNEC in variable bomb_total_energy) is determined by the difference between the total initial energy (internal + gravitational + kinetic) and the (user-supplied) sought energy of the explosion (mostly kinetic, but some thermal, but not including energy radiated away in photons). The initial energy is calculated by subroutine conservation_compute_energies in source file conservation.F90. The sought-after final energy must be specified by the user in parameter final_energy in the parameters file.

The total energy of the bomb E_{tot} is injected in the model with exponential fall off both in time and mass coordinate. The parameters bomb_tstart and bomb_tend define the start and ending time of energy injection, in the following, we will refer to them as $t^{\mathrm{b}}_{\mathrm{start}}$ and $t^{\mathrm{b}}_{\mathrm{end}}$, respectively. The integer parameter bomb_start_point defines the inner-boundary computational cell of the bomb (which we call $m^{\mathrm{b}}_{\mathrm{start}}$) and the parameter bomb_mass_spread gives the extent of the bomb in mass coordinate (in $M_\odot$), which is the difference $m^{\mathrm{b}}_{\mathrm{end}} - m^{\mathrm{b}}_{\mathrm{start}}$. Given these parameters, SNEC (in problem.F90) finds the number of grid points over which the thermal bomb is spread and writes this information out into the file info.dat int he output directory.

We chose the energy per unit time injected into the star to be given by

$$P^{\mathrm{b}}(t) = d' e^{-c't} \,. \tag{15}$$

If we require that the ratio $P_{\mathrm{bomb}}(t^{\mathrm{b}}_{\mathrm{start}})/P_{\mathrm{bomb}}(t^{\mathrm{b}}_{\mathrm{end}}) = R_{\mathrm{t}}$, where R_{t} is set by the hard-coded parameter ratio_time in source file bomb_profile.F90, then we find

$$c' = \frac{\ln R_t}{(t^{\mathrm{b}}_{\mathrm{end}} - t^{\mathrm{b}}_{\mathrm{start}})} \,, \qquad d' = \frac{c' E_{\mathrm{tot}}}{e^{-c' t^{\mathrm{b}}_{\mathrm{start}}} - e^{-c' t^{\mathrm{b}}_{\mathrm{end}}}} \,. \tag{16}$$

We choose the energy injected per unit time and per mass of the i-th cell to be of the form

$$P^{\mathrm{b}}_{\mathrm{m},i}(m_i) = b' e^{-a' m_i} \,. \tag{17}$$

If we impose the condition that the ratio $P^{\mathrm{b}}_{\mathrm{m},i}(m^{\mathrm{b}}_{\mathrm{start}})/P^{\mathrm{b}}_{\mathrm{m},i}(m^{\mathrm{b}}_{\mathrm{end}}) = R_m$, where R_m is set by the hard-coded parameter ratio_mass in source file bomb_profile.F90, then we obtain

$$a' = \frac{\ln R_m}{m^{\mathrm{b}}_{\mathrm{end}} - m^{\mathrm{b}}_{\mathrm{start}}} \,, \qquad b' = \frac{d' e^{-c't}}{\sum\limits_i e^{-a' m_i} \Delta m_{i+1/2}} \,, \tag{18}$$

where the summation in the expression for b' is performed over the cells that comprise the thermal bomb. Note that in order to find d' we perform an integration assuming a continuous time coordinate in Eq. 16. This is justified, since the timestep (at the start of the calculation) will be very small compared to the duration of energy injection by the bomb. Individual cell masses, however, will generally not be sufficiently small for a continuum approximation, so we use a sum over discrete mass cells in Eq. 18.

At the evolution equation level, the thermal bomb power (Eq. 17) is added to the right-hand side of Eq. (2) for the time interval [`bomb_tstart`,`bomb_tend`].

3 Finite-Difference Form and Evolution of the Equations

In this section, we present SNEC's numerical scheme for the implementation of Eqs. 1–3. We present the finite-difference equations in the order they are executed in `hydro_rad.F90`. The SNEC's hydrodynamics scheme is based on Mezzacappa & Bruenn 1993 [2] and was originally implemented in the pure-hydro version of SNEC, the `blcode`, which is also available from `http://stellarcollapse.org/SNEC`. SNEC's `hydro.F90` is essentially `blcode`'s hydrodynamics solver. In pure-hydro mode, SNEC still computes the luminosity of the radiation emitted from the explosion, but there is no radiation-matter coupling.

The velocity is staggered in time and kept at half timesteps ($n+1/2$, indicated as a superscript). This is a way of making the evolution second-order accurate in time (see, e.g., [2, 9]). The update of the velocity at the inner edge of cell i at time $n+1/2$ is given by

$$\begin{aligned} v_i^{n+1/2} \quad &= \quad v_i^{n-1/2} - \Delta t_v \frac{Gm_i}{(r_i^n)^2} - \Delta t_v 4\pi (r_i^n)^2 \frac{P^n_{i+1/2} - P^n_{i-1/2}}{\Delta m_i} \\ &\quad - \Delta t_v 4\pi \frac{(r^n_{i+1/2})^2 Q^{n-1/2}_{i+1/2} - (r^n_{i-1/2})^2 Q^{n-1/2}_{i-1/2}}{\Delta m_i} \ , \end{aligned} \tag{19}$$

where $v_i^{n-1/2}$ is the old velocity and the second and third terms correspond to the right-hand-side of Eq. 3. The fourth term is due to the artificial viscosity, which is needed to stabilize the numerical evolution at discontinuities (= shocks) and to avoid the development of artificial discontinuities. This is the von Neumann-Richtmyer approach to handling numerical hydrodynamics (e.g., [1, 9]). The artificial viscosity should be zero wherever the flow is smooth and non-zero where there are discontinuous changes in the state variables to damp numerical noise that will invariably arise there. We provide the detailed form of Q in §3.1. In the momentum equation (Eq. 19), the artivicial viscosity is time-lagged for additional stability [2]. The velocity update in Eq. 19 is spatially centered, thus is also second-order in space.

The timestep for the velocity update Δt_v is given as the average of the previous and the current timestep,

$$\Delta t_v \equiv \frac{1}{2}(\Delta t^{n-1/2} + \Delta t^{n+1/2}) \ , \tag{20}$$

where

$$\Delta t^{n+1/2} \equiv t^{n+1} - t^n \ . \tag{21}$$

We elaborate in §3.2 on how we chose $\Delta t^{n+1/2}$.

Once the new velocity at time $n+1/2$ is known, the radial (spatial) coordinates are updated according to

$$r_i^{n+1} = r_i^n + \Delta t^{n+1/2} v_i^{n+1/2} \; . \tag{22}$$

The cell center is defined so that the region between the inner edge of the cell and the cell center contains half the mass in a cell. The discrete radial coordinate of the cell center at time $n+1$ is computed via a volume average (assuming that the density is constant across a cell):

$$r_{i+1/2}^{n+1} = \left[\frac{(r_i^{n+1})^3 + (r_{i+1}^{n+1})^3}{2} \right]^{1/3} \; . \tag{23}$$

Next, we use the updated radial coordinates to update the cell densities,

$$\rho_{i+1/2}^{n+1} = \frac{\Delta m_{i+1/2}}{\frac{4}{3}\pi[(r_{i+1}^{n+1})^3 - (r_i^{n+1})^3]} \; . \tag{24}$$

Note that the finite difference representation of the above equation for the density is spatially centered and, hence, has a second-order truncation error in Δm.

Finally, we update the specific internal energy. The discrete form of the energy equation (Eq. 2) may be written as (for clarity, we drop the energy deposition from radioactive nickel ϵ_{Ni} here, but it is included in the code)

$$\begin{aligned} \epsilon_{i+1/2}^{n+1} \quad = \quad & \epsilon_{i+1/2}^n - \frac{1}{2}\left(P_{i+1/2}^{n+1} + P_{i+1/2}^n\right)\Delta V_{i+1/2} \\ & - \left[\theta(L_{i+1}^{n+1} - L_i^{n+1}) + (1-\theta)(L_{i+1}^n - L_i^n)\right] \frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}} \\ & -4\pi Q_{i+1/2}^{n+1/2} \left[\frac{1}{2}(r_{i+1/2}^{n+1} + r_{i+1/2}^n)\right]^2 \left(v_{i+1}^{n+1/2} - v_i^{n+1/2}\right) \frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}} \; , \end{aligned} \tag{25}$$

where $\Delta V_{i+1/2} = \left(\frac{1}{\rho_{i+1/2}^{n+1}} - \frac{1}{\rho_{i+1/2}^n}\right)$.

The luminosity L_i^n is given by

$$L_i^n \; = - \left[4\pi(r_i^n)^2\right]^2 \left(\frac{1}{\kappa}\right)_i^n \frac{ac\lambda_i^n}{3} \frac{(T_{i+1/2}^n)^4 - (T_{i-1/2}^n)^4}{\Delta m_i} \; , \tag{26}$$

where

$$\left(\frac{1}{\kappa}\right)_i^n = \frac{(T_{i+1/2}^n)^4/\kappa_{i+1/2}^n + (T_{i-1/2}^n)^4/\kappa_{i-1/2}^n}{(T_{i+1/2}^n)^4 + (T_{i-1/2}^n)^4} \; . \tag{27}$$

SNEC stores the inverse opacity $(1/\kappa)$ in the array `inv_kappa`. The ■ **[TODO: where does the flux limiter come from? Need reference and explanation]** ■ flux limiter λ_i^n is

$$\lambda_i^n = \frac{6 + 3R_i^n}{6 + 3R_i^n + (R_i^n)^2} \; , \tag{28}$$

where

$$R_i^n = \frac{8\pi(r_i^n)^2}{(T_{i+1/2}^n)^4 + (T_{i-1/2}^n)^4} \frac{|(T_{i+1/2}^n)^4 - (T_{i-1/2}^n)^4|}{\Delta m_i} \left(\frac{1}{\kappa}\right)_i^n \; . \tag{29}$$

The flux limiter, inverse opacity, and luminosity are computed in subroutine `luminosity` (in source file `luminosity.F90`) called from subroutine `hydro_rad` (and also from subroutine `hydro`, but for purely diagnostic reasons).

■ **[TODO: continue editing here]** ■ Collecting together the terms of the Eq. 25, which depend on the temperature T^{n+1}, one gets

$$\begin{aligned}
&\epsilon^{n+1}_{i+1/2} + \frac{1}{2}P^{n+1}_{i+1/2}\Delta V_{i+1/2} + \theta\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}(L^{n+1}_{i+1} - L^{n+1}_{i})\\
&\qquad = \epsilon^{n}_{i+1/2} - \frac{1}{2}P^{n}_{i+1/2}\Delta V_{i+1/2} - (1-\theta)\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}(L^{n}_{i+1} - L^{n}_{i})\\
&\qquad -4\pi Q^{n+1/2}_{i+1/2}\left[\frac{1}{2}(r^{n+1}_{i+1/2} + r^{n}_{i+1/2})\right]^2\left(v^{n+1/2}_{i+1} - v^{n+1/2}_{i}\right)\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}\,,
\end{aligned}\tag{30}$$

where $\epsilon^{n+1}_{i+1/2}$ and $P^{n+1}_{i+1/2}$ depend only on $T^{n+1}_{i+1/2}$, while $L^{n+1}_{i+1} = L^{n+1}_{i+1}(T^{n+1}_{i+3/2}, T^{n+1}_{i+1/2})$ and $L^{n+1}_{i} = L^{n+1}_{i}(T^{n+1}_{i+1/2}, T^{n+1}_{i-1/2})$. Assuming

$$T^{n+1}_{i-1/2} = T^{*}_{i-1/2} + \delta T_{i-1/2}\,, T^{n+1}_{i+1/2} = T^{*}_{i+1/2} + \delta T_{i+1/2}\,, T^{n+1}_{i+3/2} = T^{*}_{i+3/2} + \delta T_{i+3/2}\,,$$

and linearizing with respect to the small parameters $\delta T_{i-1/2}$, $\delta T_{i+1/2}$ and $\delta T_{i+3/2}$ one can rewrite Eq. 30 in the form

$$A_i\delta T_{i+3/2} + B_i\delta T_{i+1/2} + C_i\delta T_{i-1/2} = D_i\,,\tag{31}$$

where the coefficients are defined as

$$\begin{aligned}
A_i &= \theta\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}\left.\frac{\partial L^{n+1}_{i+1}}{\partial T^{n+1}_{i+3/2}}\right|_{\delta T_{i+\frac{3}{2}}\to 0}\,,\\
B_i &= \left.\frac{\partial \epsilon^{n+1}_{i+1/2}}{\partial T^{n+1}_{i+1/2}}\right|_{\delta T_{i+\frac{1}{2}}\to 0} + \frac{1}{2}\Delta V_{i+1/2}\left.\frac{\partial P^{n+1}_{i+1/2}}{\partial T^{n+1}_{i+1/2}}\right|_{\delta T_{i+\frac{1}{2}}\to 0} + \theta\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}\left.\left(\frac{\partial L^{n+1}_{i+1}}{\partial T^{n+1}_{i+1/2}} - \frac{\partial L^{n+1}_{i}}{\partial T^{n+1}_{i+1/2}}\right)\right|_{\delta T_{i+\frac{1}{2}}\to 0}\,,\\
C_i &= -\theta\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}\left.\frac{\partial L^{n+1}_{i}}{\partial T^{n+1}_{i-1/2}}\right|_{\delta T_{i-\frac{1}{2}}\to 0}\,,
\end{aligned}\tag{32}$$

and the right hand side is equal to

$$\begin{aligned}
D_i &= \epsilon^{n}_{i+1/2} - \frac{1}{2}\Delta V_{i+1/2}P^{n}_{i+1/2} - (1-\theta)\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}(L^{n}_{i+1} - L^{n}_{i})\\
&- 4\pi Q^{n+1/2}_{i+1/2}\left[\frac{1}{2}(r^{n+1}_{i+1/2} + r^{n}_{i+1/2})\right]^2\left(v^{n+1/2}_{i+1} - v^{n+1/2}_{i}\right)\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}\\
&- \left.\left[\epsilon^{n+1}_{i+1/2} + \frac{1}{2}\Delta V_{i+1/2}P^{n+1}_{i+1/2} + \theta\frac{\Delta t^{n+1/2}}{\Delta m_{i+1/2}}(L^{n+1}_{i+1} - L^{n+1}_{i})\right]\right|_{\delta T\to 0}\,.
\end{aligned}\tag{33}$$

Calculation of the coefficients (Eq. 32) and right hand side (Eq. 33) is performed in the subroutine `matrix_arrays` (in `arrays.F90`). In the current version of the code we neglect the temperature dependence of the opacity and flux-limiter and use the derivatives of the luminosity with respect to temperature in the simple form

$$\begin{aligned}
\frac{\partial L^{n+1}_{i+1}}{\partial T^{n+1}_{i+3/2}} &= -\left[4\pi (r^{n+1}_{i+1})^2\right]^2 \left(\frac{1}{\kappa}\right)^n_{i+1} \frac{ac\lambda^{n+1}_{i+1}}{3} \frac{4(T^{n+1}_{i+3/2})^3}{\Delta m_{i+1}} ,\\
\frac{\partial L^{n+1}_{i+1}}{\partial T^{n+1}_{i+1/2}} &= \left[4\pi (r^{n+1}_{i+1})^2\right]^2 \left(\frac{1}{\kappa}\right)^n_{i+1} \frac{ac\lambda^{n+1}_{i+1}}{3} \frac{4(T^{n+1}_{i+1/2})^3}{\Delta m_{i+1}} ,\\
\frac{\partial L^{n+1}_{i}}{\partial T^{n+1}_{i+1/2}} &= -\left[4\pi (r^{n+1}_{i})^2\right]^2 \left(\frac{1}{\kappa}\right)^n_{i} \frac{ac\lambda^{n+1}_{i}}{3} \frac{4(T^{n+1}_{i+1/2})^3}{\Delta m_{i}} ,\\
\frac{\partial L^{n+1}_{i}}{\partial T^{n+1}_{i-1/2}} &= \left[4\pi (r^{n+1}_{i})^2\right]^2 \left(\frac{1}{\kappa}\right)^n_{i} \frac{ac\lambda^{n+1}_{i}}{3} \frac{4(T^{n+1}_{i-1/2})^3}{\Delta m_{i}} .
\end{aligned}$$

The derivatives of ϵ and P with respect to T are obtained from the equation of state.

Taking as an initial guess $T^* = T^n$ at every grid point, we solve the obtained linear system (Eq. 31) for all grid points except the outer boundary one. To invert the tridiagonal matrix of coefficients we use the `lapack` package. After inversion the values of T^* are updated as $T^* \to T^* + \delta T$ and all the process is repeated until $\min(\delta T / T^*)$ becomes less than the chosen tolerance level `EPSTOL` (set in `hydro_rad.F90`). Note, that when evolving the energy equation we always use the value of the opacity κ at the previous (n-th) time step, while the flux limiter is recalculated at each iteration with the updated value of T^*.

As a boundary condition for the energy equation we assume that the luminosity at the surface of the star is equal to the luminosity at the closest innermost point, $L_{i_{\max}} = L_{i_{\max}-1}$ (hardcoded in `matrix_arrays` subroutine in `arrays.F90`). With this condition the luminosity terms in the Eq. 25 for the $\epsilon^{n+1}_{i_{\max}-1/2}$ cancel each other and the boundary value of temperature, $T_{i_{\max}+1/2}$, does not participate in the evolution of equations, as well as $\epsilon_{i_{\max}+1/2}$. This boundary condition describes the real physical situation when the photosphere of the models is well resolved.

3.1 Artificial Viscosity

The Euler equations admit weak solutions, i.e. solutions for which the integral form of the conservation laws hold, but the differential form is violated, because of discontinuities in the state variables. As a consequence, a naive finite-difference treatment of the Euler equations will lead to the growth and eventual blow-up of oscillations near discontinuities (e.g., shocks, contact discontinuities, surfaces of stars etc.). The idea of von Neumann & Richtmyer [1] is to avoid such oscillations by artificially spreading out discontinuities over multiple grid cells and thus avoiding the development of unstable oscillations. The *artificial viscosity* used for this should be non-zero only at discontinuities and zero everywhere else.

Mezzacappa & Bruenn [2] use an artificial viscosity prescription optimized for collapse. However, in SNEC, we use the simpler original von Neumann & Richtmyer form,

$$Q \equiv \begin{cases} c_0^2 \rho (\partial v/\partial k)^2 & \partial v/\partial k < 0 \\ 0 & \text{otherwise} , \end{cases} \tag{34}$$

where $\partial/\partial k$ denotes the deriviate with respect to an integer Lagrangian coordinate (some cell index k). The finite-difference form of this implemented in `artificial_viscosity.F90` is

$$Q^{n+1/2}_{i+1/2} = \begin{cases} c_0^2 \rho^{n+1}_{i+1/2} (v^{n+1/2}_{i+1} - v^{n+1/2}_{i})^2 & \text{if } (v^{n+1/2}_{i+1} - v^{n+1/2}_{i}) < 0 \\ 0 & \text{otherwise .} \end{cases} \tag{35}$$

We set $c_0^2 = 2$, following [2, 6].

3.2 Timestep

The timestep used in the update of the hydrodynamic equations must not violate causality, that is, it must be no larger than the time it takes a sound wave to travel across a grid cell. In addition to the causality constraint, the timestep may further need to be limited to ensure stability of the numerical implementation (see, e.g., [1, 3, 9]).

In SNEC, the timestep is determined in `timestep.F90` according to the following prescription. We first compute a local timestep for each cell i,

$$\Delta t^{n+1/2}_{i+1/2} = \min \left(\frac{r^n_{i+1} - r^n_i}{|v^{n-1/2}_i + c^n_{s,i+1/2}|}, \frac{r^n_{i+1} - r^n_i}{|v^{n-1/2}_i - c^n_{s,i+1/2}|} \right) , \tag{36}$$

where c_s is the speed of sound obtained from the EOS. We then use the global minimum as the new timestep,

$$\Delta t^{n+1/2} = CFL \min(\{t^{n+1/2}_i\}) , \tag{37}$$

where CFL is the Courant-Friedrichs-Lewy factor, which we set to 0.95 for stability.

3.3 Rosseland mean opacity

The subroutine `opacity` finds opacity κ in each grid point of the model from the existing tables of the Rosseland mean opacities for different compositions, temperatures and densities of matter. The output of the subroutine consists of the arrays `kappa_table`, the opacity found from the tables, and `kappa`, the same opacity after taking into account the opacity floor. The values of `kappa` participate in the evolution of the hydrodynamics equations, while the values of `kappa_table` are used while looking for the position of the photosphere of the star, and for defining the observed luminosity (see `analysis.F90`). The derivative of the opacity with respect to temperature (array `dkappadt`) is outputted as well, but does not participate in the evolution of the equations in the current version of the code.

In the high temperature regime (from $\log T \sim 3.75$ to $\log T = 8.7$), we use the set of OPAL tables of Type II with the nominal metallicity $Z_{\text{tab}} = 0.02$ (equal to the solar metallicity), together with the interpolating routine, taken from the OPAL website `http://opalopacity.llnl.gov/opal.html` (contained in the file `opal_opacity.F90`). For the details of the generation of these tables, see [10]. The routine, provided on the OPAL website, performs interpolation between five variables, namely,

X ,	hydrogen mass fraction,
δX_{C} ,	enhanced carbon mass fraction,
δX_{O} ,	enhanced oxygen mass fraction,
T_6 ,	temperature in millions of degrees Kelvin,
$R = \rho / T_6^3$,	where ρ is the density of matter.

In the code the four last variables are represented by `xxc`, `xxo`, `t6` and `R_op`, respectively. The enhanced mass fractions in i-th grid point are found as

$$\begin{aligned}\delta X_{i,\mathrm{C}} &= \max\left(0,\ X_{i,\mathrm{C}} - Z_{\mathrm{tab}} X_{\mathrm{C}}^{\mathrm{sol}}\right)\ ,\\ \delta X_{i,\mathrm{O}} &= \max\left(0,\ X_{i,\mathrm{O}} - Z_{\mathrm{tab}} X_{\mathrm{O}}^{\mathrm{sol}}\right)\ ,\end{aligned}$$

where $X_{i,\mathrm{C}}$ and $X_{i,\mathrm{O}}$ are the total mass fractions of carbon/oxygen in the considered point, and $X_{\mathrm{C}}^{\mathrm{sol}}$ and $X_{\mathrm{O}}^{\mathrm{sol}}$ are the relative mass fractions of carbon/oxygen in the solar composition, equal to 0.173285 and 0.482273, respectively (see the abundances inside the codata files). Note that we do not use the option of smoothing of the original tabular data, having `ismdata=1` in the interpolating routine (see the explanation in `opal_opacity.F90`).

The interpolating routine is called in cases, in which the values of temperature and density in the considered point lie within the range of the opacity tables. Otherwise, the opacity in each subsequent grid point dublicates the opacity in the previous point (typically, the values of density start to go out of the table range in very rarified outer layers of the expanding model, in this case the described procedure just flatens the opacity in the outer region).

After the tabular value of opacity is found, we choose the maximum between this value and the value of the opacity floor in the considered point. The opacity floor is set up at the beginning of the simulation in `problem.F90`. The value of the opacity floor in i-th grid point is chosen to be proportional to the metallicity Z_i in this point. In addition we require the values of the opacity floor in the envelope (solar metallicity $Z_\odot = 0.02$) and core ($Z = 1$) to be equal to $\mathrm{OF}_{\mathrm{env}} = 0.01$ and $\mathrm{OF}_{\mathrm{core}} = 0.24$, respectively, so that

$$\mathrm{OF}_i = \frac{\mathrm{OF}_{\mathrm{core}} Z_\odot - \mathrm{OF}_{\mathrm{env}} + Z_i(\mathrm{OF}_{\mathrm{env}} - \mathrm{OF}_{\mathrm{core}})}{Z_\odot - 1}\ . \tag{38}$$

The specific chosen values of $\mathrm{OF}_{\mathrm{env}}$ and $\mathrm{OF}_{\mathrm{core}}$ are based on the recent calibration of the results of the LTE code against the results of the multifrequency code, performed in [6].

In the low temperature regime (down to $\log T = 2.7$), the OPAL opacity tables are supplemented by the tables presented in [11]. These tables exist for the composition mixtures with zero enhanced carbon/oxygen mass fractions, $\delta X_{i,\mathrm{C}} = 0$ and $\delta X_{i,\mathrm{O}} = 0$. For these cases we use the values from the low temperature tables starting from their upper temperature boundary $\log T = 4.5$. We don't interpolate between the tables in the overlap region ($3.75 < \log T < 4.5$), giving the preference to the low temperature tables, because they take into account the contribution of molecular lines, which are not included in the OPAL tables.[2]

For the tables with nonzero enhanced carbon and oxygen mass fractions we simply fill in the regions, where the data is absent, by the last meaningful values from right to left (from higher to lower densities) and then from bottom up (from higher to lower temperatures). This is done in order to have the equal size of all tables and use the same OPAL interpolation routine (note that due to this reason we changed the parts of the interpolation routine, where the size and structure of the tables were encoded). The order in which the filling is performed is motivated by the fact that the opacity generally demonstrates a stronger dependence on the temperature than on the density in the considered region of the parameter space.

[2] The same low temperature table for zero hydrogen mass fraction is used in codataa and codatab files.

Although generation of the low temperature tables, exactly supplementing the OPAL Type II tables, is technically possible with the existing codes, it would not be very helpful from the physical point of view. The reason is that the range of change of the carbon and oxygen mass fractions in the OPAL tables is very large and relatively sparsely covered by the tables. For the high temperature regime (mostly determined by the atomic lines) the opacity is expected to change smoothly and the interpolation between the different tables may be reliable. For the low temperatures, when the opacity is mostly determined by the molecular lines, the opacity is not expected to change smoothly between the different compositional mixtures of the OPAL tables, and one cannot rely on the smooth interpolation between them (see [12]).

It should be noted that the opacity, calculated as described above, is expected to be systematically higher than the correct value (because physically we expect the opacity in carbon/oxygen enhanced regions to go down with the decreasing temperature, instead of staying constant). However, the low temperature region of the tables lies below the currently used opacity floor, and uncertainty in the tables almost does not influence the evolution, having noticeable effect only on the position of the photosphere in the carbon/oxygen rich regions of the star. Therefore, it affects the transition region between plateau and the nickel part of the light curve, which cannot be reproduced very accurately already due to the LTE assumption, adopted in the code.

3.4 Boundary Conditions

We must specify boundary conditions and some hydrodynamic/thermodynamic state variables at both the inner and the outer boundaries of our grid. The default boundary conditions in SNEC are the following. The user should feel free to experiment and change them according to their need.

Inner Boundary, `i=1` (quantities not specified are computed and not fixed)

Quantity	Description/Notes	Source File
$v_1 = 0$	Set to v_{piston} only when piston is active.	`hydro.F90, hydro_rad.F90`
$L_1 = 0$	Luminosity at the inner boundary	`arrays.F90`

Outer Boundary, `i=imax` (quantities not specified are computed and not fixed)

Quantity	Description/Notes	Source File
$Q_{\text{imax}} = 0$	Artificial viscosity	`artificial_viscosity.F90`
$\rho_{\text{imax}+1/2} = 0$	Density	`read_profile.F90, hydro.F90, hydro_rad.F90`
$\epsilon_{\text{imax}+1/2} = 0$	Spec. int. energy.	`read_profile.F90, hydro.F90, hydro_rad.F90`
$T_{\text{imax}+1/2} = 0$	Temperature	`read_profile.F90, hydro.F90, hydro_rad.F90`
$P_{\text{imax}+1/2} = 0$	Pressure	`read_profile.F90, hydro.F90, hydro_rad.F90`
$L_{\text{imax}} = L_{\text{imax}-1}$	Luminosity	`arrays.F90`

Note that $P_{\text{imax}+1/2}$, L_{imax} and Q_{imax} are the only important boundary conditions, since the other quantities at `imax` and `imax+1/2` are never used in the evolution.

4 Heating by the radioactive decay of ^{56}Ni

The subroutine `nickel.F90` is called once in a chosen interval of time, given by the parameter `Ni_period`. It is based on the work [13] and solves the transfer equation for the γ-rays in gray approximation, assuming that the γ-ray opacity, $\kappa_\gamma = 0.06 Y_e$ cm^2 g^{-1}, where Y_e is the electron fraction, is purely absorptive and independent of energy.

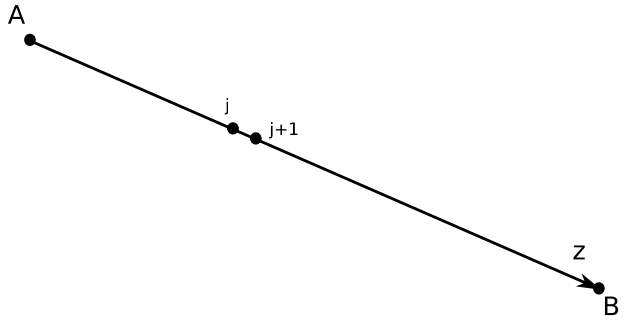

Figure 1: Illustration to the Eqs.(39)-(44).

Consider a γ-ray, propagating along the axis z from the point A to the point B (see the scheme in the Fig.1). The transfer equation for the energy-integrated intensity I may be written as

$$\frac{\partial I}{\partial z} = \eta - \kappa_\gamma \rho I \,, \tag{39}$$

where ρ is the density of matter, $\eta = X_{\rm Ni}\epsilon_{\rm rad}\rho/4\pi$ is the local total γ-ray emissivity, $X_{\rm Ni}$ is the mass fraction of ^{56}Ni. The time dependent rate of energy release per gram of radioactive nickel, $\epsilon_{\rm rad}$, is equal to

$$\begin{aligned}\epsilon_{\rm rad} &= 3.9\times 10^{10}\exp(-t/\tau_{\rm Ni})\\ &\quad +6.78\times 10^{9}\left[\exp(-t/\tau_{\rm Co}) - \exp(-t/\tau_{\rm Ni})\right] \mathrm{erg\,g^{-1}\,s^{-1}}\,,\end{aligned} \tag{40}$$

where $\tau_{\rm Ni}$ and $\tau_{\rm Co}$ are the mean lifetimes of ^{56}Ni and ^{56}Co, equal to 8.8 and 113.6 days, respectively. Introducing the optical depth along the ray

$$d\tau = -\kappa_\gamma \rho dz \,, \tag{41}$$

and defining $I' = (4\pi\kappa_\gamma/\epsilon_{\rm rad})I$, one can rewrite Eq. 39 (following [13]) as

$$\frac{dI'}{d\tau} = I' - X_{\rm Ni} \,. \tag{42}$$

Integrating it one can get

$$I'(\tau_{j+1}) = I'(\tau_j)e^{-(\tau_j-\tau_{j+1})} + \int_{\tau_{j+1}}^{\tau_j} X_{\rm Ni}(\tau)e^{-(t-\tau_{j+1})}dt \,, \tag{43}$$

where the point j and $j+1$ are as indicated in the Fig.1 (note that $\tau_{j+1} < \tau_j$). Assuming that $X_{\rm Ni}(\tau) = X_{{\rm Ni},j}$ is constant between these points, one can perform the integration to get

$$I'(\tau_{j+1}) = I'(\tau_j)e^{-(\tau_j-\tau_{j+1})} + X_{{\rm Ni},j}\left[1 - e^{-(\tau_j-\tau_{j+1})}\right] \,. \tag{44}$$

The rate at which the energy is deposited in the given grid point is equal to $\epsilon_{\rm rad}d$, where the deposition function d is found as an integral of I' over all solid angles

$$d = \frac{1}{4\pi}\oint I' d\omega \,. \tag{45}$$

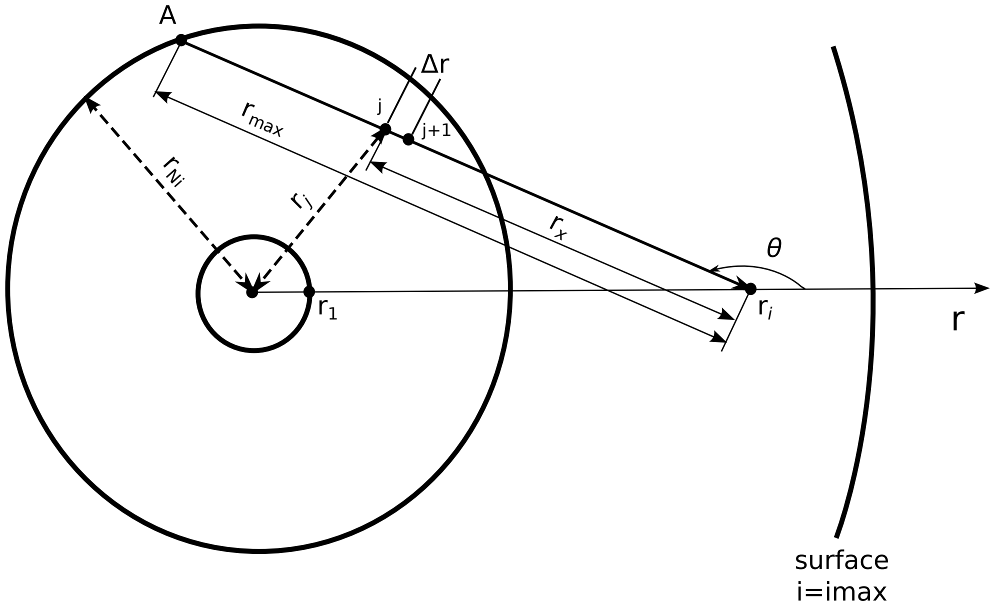

Figure 2: Geometrical scheme behind the calculations, performed in the subroutine.

Fig.2 shows the geometrical scheme, used for the calculations performed in the subroutine (with the labeling matching the names of variables in the code). $r_{\rm Ni}$ is the radius of the sphere, out of which the mass fraction of ^{56}Ni is less than chosen minimum mass fraction, given by the local parameter `NIMIN`. Although using `NIMIN` introduces a certain (controlable) error in the calculations, without it the numerical integration described below would be highly inefficient for the cases, when ^{56}Ni is concentrated close to the center.

In order to find the deposition function d_i in the i-th grid point with the radial coordinate r_i we iteratively evolve Eq. 44 along a set of rays, corresponding to the different values of the polar angle θ. For each ray we start from the outermost point of each ray, having non-negligible mass fraction of ^{56}Ni, finish the iterations at the point r_i, and perform the angular averaging according to Eq. 45.

In particular, for the ray, shown in Fig.2, we evolve Eq. 44 starting from the point `A` and use as a boundary condition $I' = 0$ at this point (no external sources of γ-rays outside the radius $r_{\rm Ni}$). The distance $r_{\rm max}$ between `A` and r_i is discretized with the constant radial increment $\Delta r = r_{\rm max}/N_r$, where N_r is the number of points per radial integral (given by the local parameter `npoints_radial_integration` in the code). Note that we use a fixed amount of points N_r per length $r_{\rm max}$ instead of the fixed increment Δr, because the $r_{\rm max}$ will strongly vary for the different grid points, and taking into account that the envelope of the exploded star will continue to expand, it is difficult to choose a unique suitable Δr.

The following geometrical relations hold between the different distances in the Fig.2

$$
\begin{aligned}
r_{\rm max} &= -r_i\cos\theta + \sqrt{r_i^2\cos^2\theta - (r_i^2 - r_{\rm Ni}^2)} \\
r_j^2 &= r_i^2 + r_x^2 + 2r_i r_x\cos\theta \, .
\end{aligned}
\tag{46}
$$

Variables `rho_j`, `kappa_gamma_j` and `comp_Ni_j` in the code stand for the values of ρ, κ_γ and $X_{\rm Ni}$, respectively, at the distance r_j from the center (found by linear interpolation in radial coordinate, using the `map_map` routine). Note, that the density in the excised region of the profile, $r < r_1$, is assumed to be zero, because the compact remnant neutron star will occupy only very small central region of this space.

Having found the value of I' at the i-th grid point due to the ray defined by the angle θ, $I'_i(\theta)$, we need to integrate it over the solid angle $d\omega = \sin\theta d\theta d\phi$, where ϕ is the azimuthal angle. In case when $r_i < r_{\rm Ni}$, the angular integration is performed between $\theta = \pi$ and $\theta = 0$, while in case when $r_i > r_{\rm Ni}$, the integration is performed between $\theta = \pi$ and $\theta = \theta_{{\rm min},i}$ (`th_min(i)` in the code), obtained from the relation

$$\cos\theta_{{\rm min},i} = -\frac{\sqrt{r_i^2 - r_{\rm Ni}^2}}{r_i} \,. \tag{47}$$

Integration is performed using the simple rectangle method with the angular increment $\Delta\theta = (\pi - \theta_{{\rm min},i})/N_\theta$, where N_θ is the number of points per angular integral (given by local parameter `npoints_angular_integration` in the routine).

Taking into account the azimuthal symmetry, the deposition function in the i-th grid point, being equal to the angular average of I', is given by

$$d_i = \frac{1}{2}\sum_\theta I'_i(\theta)\sin\theta\Delta\theta \,, \tag{48}$$
$$\theta = \pi, \pi - \Delta\theta, ..., \theta_{{\rm min},i} + \Delta\theta \,.$$

Multiplied on $\epsilon_{\rm rad}$, this value gives the amount of energy per unit mass per unit time, deposited in the i-th grid point (`Ni_heating(i)` in the code). The subroute also calculates the total amount of energy per second deposited by γ-rays (`Ni_total_luminosity` in the code).

5 Equations of State

An equation of state (EOS) is crucial for closing the system of hydrodynamics equations. At the moment SNEC provides a choice between two EOS, but we plan to add more options in the future. The top-level `eos` routine, located in `eos_switch.F90`, switches between the EOS and calles lower-level routines, located in `eos_content.F90`. Some EOS parameters are set in `problem.F90`.

The EOS is selected by the user with the `eoskey` variable that is set in the `parameters` file. In the following sections we summarize the EOS options.

5.1 Ideal single-particle Boltzmann Gas EOS (`eoskey = 1`)

In this case pressure P, specific internal energy ϵ and speed of sound c_s are given by simple expressions

$$P = k_{\rm B}N_{\rm A}\rho T \,, \qquad \epsilon = \frac{k_{\rm B}N_{\rm A}T}{\Gamma - 1} \,, \qquad c_s^2 = \Gamma k_{\rm B}N_{\rm A}T \,, \tag{49}$$

where $N_{\rm A}$ is the Avogadro's constant and Γ is the ratio of specific heats $\Gamma \equiv c_p/c_V$ chosen to be 1.4 in `problem.F90`.

This EOS is not recommended for the light curve generation and its main purpose is to demonstrate the ability of the `hydro.F90` routine to pass the Sedov blastwave test. For that purpose one may use the profile files `sedov.short` and `sedov.iso.dat` (in `profile` folder), corresponding to a sphere of radius $R = 2$ cm and constant density $\rho_0 = 1$ g cm^{-3}, consisting of only neutrons. Parameter file `parameters_sedov` is tuned for the explosion of this model with energy 1 erg (without including radiation into the scheme, with gravity turned off).

Analytical solution of this problem with the same setup is generated by script `sedov.py`, based on the work of James R. Kamm [14]. The script outputs density as a function of radial coordinate, which can be compared to the SNEC's `rho.xg` and `radius.xg` output files.

When performing this Sedov blastwave test one may notice that the shock radius given by SNEC is slightly larger than the analytic solution. This reflects the imperfection of the bomb setup and can be seen in the last column of the `conservation.dat` output file, giving the total energy of the system minus its initial energy as a function of time. One can see that for the chosen parameters of thermal bomb the energy, actually injected into the system, is few percent larger than the nominal explosion energy of 1 erg. Decreasing the duration time of the thermal bomb will make the discrepancy worse. Changing the parameter `final_energy` from 1 erg to 0.975 erg, one can reproduce the analytic solution more precisely and see from `conservation.dat` that the actual energy given to the system in this case is very close to 1 erg. This demonstrates that the current setup of the thermal bomb works with the accuracy of a few percent.

5.2 Paczynski EOS (`eoskey = 2`)

This is a simplified analytic EOS for a mixture of ions, photons, and semi-degenerate/semi-relativistic electrons. It is based on the work of Paczynski [15], with the addition of corrections due to the partial ionization. To calculate these corrections we closely follow paragraph 9.18 of [16].

The total pressure is calculated as a sum of contributions from the ions, electrons and radiation

$$P = P_{\rm ion} + P_{\rm e} + P_{\rm rad} \,. \tag{50}$$

The ion part is equal to

$$P_{\rm ion} = N\rho k_{\rm B} T \,, \tag{51}$$

where

$$N = \frac{1}{m_{\rm p}} \sum_k \frac{X_k}{A_k} \,, \tag{52}$$

is the number of heavy particles per unit mass, $m_{\rm p}$ is the mass of proton, A_k is the mass number of the k-th element. The radiation part of the pressure is equal to

$$P_{\rm rad} = \frac{aT^4}{3} \,, \tag{53}$$

and the electron part is given by the expressions

$$P_{\rm e} = \sqrt{P_{\rm end}^2 + P_{\rm ed}^2} \,, \qquad P_{\rm ed} = \left(P_{\rm ednr}^{-2} + P_{\rm edr}^{-2}\right)^{-1/2} \,,$$
$$P_{\rm end} = \bar{y} N \rho k_{\rm B} T \,, \quad P_{\rm ednr} = 9.91 \times 10^{12} (Y_e \rho)^{5/3} \,, \quad P_{\rm edr} = 1.231 \times 10^{15} (Y_e \rho)^{4/3} \,. \tag{54}$$

Here Y_e is the electron fraction, and $\bar{y}$ is the mean degree of ionization, equal to the ratio between the number of free ionization electrons and the total number of atoms and ions

$$\bar{y} = \sum_k \nu_k \left(\sum_s s y_s^k \right) \,, \tag{55}$$

where ν_k is the number abundance of k-th element and y_s^k is the degree of s-th ionization of the k-th element. P_{end} and P_{ed} denote the pressure of a non-degenerate and degenerate electron gas, respectively. P_{ednr} and P_{edr} correspond to the degenerate non-relativistic and degenerate relativistic cases.

Total specific internal energy contains contributions from ions, electrons and radiation, including the ionization correction as follows

$$\epsilon = \frac{3}{2}Nk_{\text{B}}T + \frac{1}{f-1}\frac{P_{\text{e}}}{\rho} + \frac{aT^4}{\rho} + N\left\{\sum_k \nu_k \left[\sum_s y_s^k \left(\sum_{m=1}^{s} \chi_{m-1}^k\right)\right]\right\} , \tag{56}$$

where f is given by

$$f = \frac{d\log P_{\text{ed}}}{d\log\rho} = \frac{5}{3}\left(\frac{P_{\text{ed}}}{P_{\text{ednr}}}\right)^2 + \frac{4}{3}\left(\frac{P_{\text{ed}}}{P_{\text{edr}}}\right)^2 . \tag{57}$$

The purpose of f is to provide correct values of the numerical prefactor in the case of degenerate and non-degenerate electron gas.

The following assumptions are adopted:

- We assume complete ionization when degenerate electron pressures are important. This is seen from the expressions for P_{ednr} and P_{edr}, which do not depend on the degree of ionization $\bar{y}$. Effects of partial ionization are taken into account only for the non-degenerate electron gas. Stellar models, which we use to generate the light curves, typically have degenerate regions close to the center, but after explosion these regions very quickly become non-degenerate. Thus, possible inaccuracy of the current assumption may influence only the very onset of the explosion, if at all.

  As a consequence of this assumption, we use the Saha equations appropriate for the non-degenerate limit when looking for the derivatives $(\partial E/\partial T)_\rho$, $(\partial P/\partial T)_\rho$ and $(\partial P/\partial \rho)_T$ below.

- Following [16], we assume that for a given density and temperature each element undergoes transition only between two neighboring ionization states, say, $r-1$-th and r-th, so that

  $$y_{r-1}^k \approx 1 - y_r^k , \qquad dy_{r-1}^k \approx -dy_r^k , \tag{58}$$

  and

  $$d\bar{y} = \sum_k \nu_k dy_r^k . \tag{59}$$

  Note, that we don't use the assumption that only one element at a time goes through the change of the ionization state, but consider contributions from all the elements.

Using these two assumptions, one can get the differentials for the pressure and energy as

$$dP = \left[Nk_{\text{B}}T + \frac{\partial P_{\text{e}}}{\partial\rho}\right]d\rho + \frac{P_{\text{ion}}}{T}dT + \frac{4aT^3}{3}dT + \frac{P_{\text{end}}^2}{P_{\text{e}}T}dT + \frac{P_{\text{end}}^2}{P_{\text{e}}\bar{y}}\sum_k \nu_k dy_r^k , \tag{60}$$

$$\begin{aligned} d\epsilon = {} & \frac{3}{2}Nk_{\text{B}}dT + N\sum_k \nu_k \chi_{r-1}^k dy_r^k + \frac{4aT^3}{\rho}dT - \frac{aT^4}{\rho^2}d\rho \\ & + \frac{\partial}{\partial\rho}\left(\frac{1}{f-1}\frac{P_{\text{e}}}{\rho}\right)d\rho + \frac{1}{f-1}\frac{P_{\text{end}}^2}{P_{\text{e}}\rho T}dT + \frac{1}{f-1}\frac{P_{\text{end}}^2}{P_{\text{e}}\rho\bar{y}}\sum_k \nu_k dy_r^k . \end{aligned} \tag{61}$$

Fraction of atoms of the k-th element in the r-th ionization state may be obtained from the Saha equation, written in the form

$$\frac{y_r^k}{1-y_r^k}\bar{y}N\rho = \frac{2g_{k,r}}{g_{k,r-1}}\frac{(2\pi m)^{3/2}}{h^3}(k_\mathrm{B}T)^{3/2}e^{-\frac{\chi_{r-1}^k}{k_\mathrm{B}T}} \, . \tag{62}$$

Taking differential of both sides of Eq. 62, one obtains

$$\frac{dy_r^k}{y_r^k(1-y_r^k)} + \frac{d\bar{y}}{\bar{y}} = -\frac{d\rho}{\rho} + \left[\frac{3}{2}\frac{1}{T} + \frac{\chi_{r-1}^k}{k_\mathrm{B}T^2}\right]dT \, . \tag{63}$$

Multiplying both sides of Eq. 63 on ν_k and summarizing across the elements, one can get

$$\sum_k \nu_k dy_r^k \left[1 + \frac{\Sigma_1}{\bar{y}}\right] = -\frac{d\rho}{\rho}\Sigma_1 + \frac{3}{2}\frac{dT}{T}\Sigma_1 + \frac{\Sigma_2}{k_\mathrm{B}T^2}dT \, , \tag{64}$$

where

$$\Sigma_1 = \sum_k \nu_k y_r^k(1-y_r^k) \, , \qquad \Sigma_2 = \sum_k \nu_k \chi_{r-1}^k y_r^k (1-y_r^k) \, . \tag{65}$$

Multiplying both sides of Eq. 63 on $\nu_k\chi_{r-1}^k$ and summarizing across the elements, one can get

$$\sum_k \nu_k \chi_{r-1}^k dy_r^k = -\frac{\Sigma_2\bar{y}}{\bar{y}+\Sigma_1}\frac{d\rho}{\rho} + \frac{3}{2}\frac{\Sigma_2\bar{y}}{\bar{y}+\Sigma_1}\frac{dT}{T} - \frac{\Sigma_2^2}{\bar{y}+\Sigma_1}\frac{1}{k_\mathrm{B}T^2}dT + \frac{\Sigma_3}{k_\mathrm{B}T^2}dT \, , \tag{66}$$

where

$$\Sigma_3 = \sum_k \nu_k (\chi_{r-1}^k)^2 y_r^k (1-y_r^k) \, . \tag{67}$$

Substituting Eqs. 64 and 66 in Eq. 60, one finally obtains

$$\begin{aligned}
\left(\frac{\partial E}{\partial T}\right)_\rho &= \frac{3}{2}Nk_\mathrm{B} + \frac{4aT^3}{\rho} + \frac{1}{f-1}\frac{P_\mathrm{end}^2}{P_\mathrm{e}\rho T}\left[1 + \frac{1}{\Sigma_1+\bar{y}}\left(\frac{3}{2}\Sigma_1 + \frac{\Sigma_2}{k_\mathrm{B}T}\right)\right] \\
&\quad + \frac{N}{T}\left[\frac{\Sigma_2}{\Sigma_1+\bar{y}}\left(\frac{3}{2}\bar{y} - \frac{\Sigma_2}{k_\mathrm{B}T}\right) + \frac{\Sigma_3}{k_\mathrm{B}T}\right] \, , && (68) \\
\left(\frac{\partial P}{\partial T}\right)_\rho &= Nk_\mathrm{B}\rho + \frac{4aT^3}{3} + \frac{P_\mathrm{end}^2}{P_\mathrm{e}T}\left[1 + \frac{1}{\Sigma_1+\bar{y}}\left(\frac{3}{2}\Sigma_1 + \frac{\Sigma_2}{k_\mathrm{B}T}\right)\right] \, , && (69) \\
\left(\frac{\partial P}{\partial \rho}\right)_T &= Nk_\mathrm{B}T - \frac{P_\mathrm{end}^2}{P_\mathrm{e}\rho}\frac{\Sigma_1}{\Sigma_1+\bar{y}} + \frac{1}{P_\mathrm{e}}\left(\frac{P_\mathrm{end}^2}{\rho} + f\frac{P_\mathrm{ed}^2}{\rho}\right) \, . && (70)
\end{aligned}$$

The speed of sound is found as

$$c_s^2 = \frac{\Gamma_1 P}{\rho} \, , \tag{71}$$

with the adiabatic exponent Γ_1 constructed from the above quantities as

$$\Gamma_1 = \frac{\chi_T^2 P}{c_V \rho T} + \chi_\rho \, , \tag{72}$$

where

$$\chi_\rho = \frac{\rho}{P}\left(\frac{\partial P}{\partial \rho}\right)_T \, , \qquad \chi_T = \frac{T}{P}\left(\frac{\partial P}{\partial T}\right)_\rho \, , \qquad c_V = \left(\frac{\partial E}{\partial T}\right)_\rho \, . \tag{73}$$

6 Saha Equations

The subroutine simple_saha.F90 solves the Saha-Boltzmann equations (in the non-degenerate limit), applicable for the description of the recombination wave propagating in the low-density envelope of the expanding star. The calculations are based on the work of Zaghloul et al. [17]. The routine finds the fractions of atoms in different ionization states for a certain number of elements from the composition profile (given by the parameter saha_ncomps), other elements are assumed to be fully ionized.

The equations below describe one element with the number density of the heavy particles n_k (denoted as n_k in the code)

$$n_k = \frac{\rho X_k}{m_p A_k} \, , \tag{74}$$

where ρ is the density, X_k is the mass fraction of the element at the considered point, m_p is the mass of proton, A_k is the mass number of the element.

In this case, the Saha-Boltzmann equations have the form

$$\frac{n_{s+1} n_e}{n_s} = 2 \frac{g_{s+1}}{g_s} \left[\frac{2\pi m_e k_\mathrm{B} T}{h^2} \right] \exp\left(-\frac{\chi_s}{k_\mathrm{B} T} \right) \, , \qquad s = 1, 2, ..., \mathbb{Z} \, , \tag{75}$$

where n_s is the number density of the atoms in the s-th ionization state[3], g_s is the statistical weigth of the s-th state, m_e is the electron's rest mass, χ_s is the ionization energy for the ionization process $s \to (s+1)$, n_e is the electron number density, $\mathbb{Z}$ is the atomic number of the considered element (not to be confused with the metallicity Z), h is Planck's constant and k_B is Boltzmann's constant. Please, note that for brevity we omit the element index k of the quantities n_s, n_e, g_s, χ_s and $\mathbb{Z}$.

Together with the condition of electro-neutrality

$$\sum_{s=2}^{\mathbb{Z}+1} (s-1) n_s = n_e \, , \tag{76}$$

and conservation of heavy nuclei

$$\sum_{s=1}^{\mathbb{Z}+1} n_s = n_k = \text{constant} \, , \tag{77}$$

these equations constitute a closed system of nonlinear algebraic equations.

Denoting the fractions of the atoms in the s-th ionization state with $y_s = n_s / n_k$ and introducing the average charge $\bar{\mathbb{Z}} = n_e / n_k$ one can rewrite the Saha-Boltzmann equations, as well as the

[3] Note, that in the original work of Zaghloul et al. (2000) the neutral state of the atom has index 0, while in this description the neutral state has index 1, in order to follow the indexing of the subroutine, where all arrays start from the index 1.

conditions of electro-neutrality and conservation of heavy nuclei in the following form

$$\sum_{s=1}^{\mathbb{Z}+1} y_s = 1\,, \tag{78}$$

$$\sum_{s=2}^{\mathbb{Z}+1} (s-1) y_s = \bar{\mathbb{Z}}\,, \tag{79}$$

$$\frac{y_{s+1}\bar{\mathbb{Z}}n_k}{y_s} = 2\frac{g_{s+1}}{g_s}\left[\frac{2\pi m_e k_\mathrm{B} T}{h^2}\right]^{3/2} \exp\left(-\frac{\chi_s}{k_\mathrm{B}T}\right) = f_s\,, \tag{80}$$
$$s = 1, 2, ...\mathbb{Z}\,.$$

From the last equation the reccurence relation follows as

$$y_{s+1} = y_s \frac{f_s}{\bar{\mathbb{Z}}n_k}\,. \tag{81}$$

The ionization energy and statistical weight are returned by the functions `ionpot` and `stat_weight`, respectively, located in `saha_tables.F90`. These functions are taken from the `Timmes EOS with Saha ionization` (see `http://cococubed.asu.edu/code_pages/eos_ionize.shtml`), and distributed with SNEC by permission of Frank Timmes (for reference, see [18, 19]).

The subroutine `simple_saha.F90` finds the factors f_s according to the Eq. 80. After that, it finds the indices of the lowest and highest ionization states with the non-negligible number densities (s_min and s_max here, `min_state` and `max_state` in the code), based on the value of $f_s/(\bar{\mathbb{Z}}n_k)$ and some chosen tolerance level (`zavtol` in the code). For example, if $f_{s'} > \mathtt{zavtol}^{-1}$ the subroutine does not solve for the s'-th ionization state, assuming $y_s = 0$ for $s = 1, ..., s'$. Accordingly, if $f_{s'} < \mathtt{zavtol}$ the subroutine does not solve for the $(s'+1)$-th ionization state, assuming $y_s = 0$ for $s = s'+1, ..., \mathbb{Z}+1$. This helps avoid numerical overflows and reduces the computational cost of the calculations.

From the Eqs. 81 and 79 one gets the fraction of atoms in the lowest considered state as

$$y_{s_\mathrm{min}} = \bar{\mathbb{Z}}\left[s_\mathrm{min} - 1 + \sum_{s=s_\mathrm{min}}^{s_\mathrm{max}-1}\left(s\prod_{j=s_\mathrm{min}}^{s}\frac{f_j}{\bar{\mathbb{Z}}n_k}\right)\right]^{-1}\,. \tag{82}$$

Combining Eqs. 82, 81 and 78, one gets the following transcendental equation for $\bar{\mathbb{Z}}$

$$F(\bar{\mathbb{Z}}) \equiv 1 - \bar{\mathbb{Z}}\left[s_\mathrm{min} - 1 + \sum_{s=s_\mathrm{min}}^{s_\mathrm{max}-1}\left(s\prod_{j=s_\mathrm{min}}^{s}\frac{f_j}{\bar{\mathbb{Z}}n_k}\right)\right]^{-1}\left[1 + \sum_{s=s_\mathrm{min}}^{s_\mathrm{max}-1}\left(\prod_{j=s_\mathrm{min}}^{s}\frac{f_j}{\bar{\mathbb{Z}}n_k}\right)\right]\,. \tag{83}$$

In the limit $s_\mathrm{min} = 1$, $s_\mathrm{max} = \mathbb{Z}+1$ this equation coincides with the Eq.(9) of [17].

In the cases when more than one ionization state has non-negligible number density, Eq. 83 is solved iteratively using the Newthon-Raphson method. The derivative $F'(\bar{\mathbb{Z}})$, needed in this case, can be written as

$$F'(\bar{\mathbb{Z}}) = \left\{\Sigma_2 - [1+\Sigma_0]\left[1 + \Sigma_3(s_\mathrm{min} - 1 + \Sigma_1)^{-1}\right]\right\}(s_\mathrm{min} - 1 + \Sigma_1)^{-1}\,, \tag{84}$$

where the following notations are adopted (with the parallel notations in the code)

$$\begin{aligned}
\Sigma_0 &= \sum_{s=s_{\min}}^{s_{\max}-1} \left(\prod_{j=s_{\min}}^{s} \frac{f_j}{\bar{\mathbb{Z}} n_k} \right) , \\
\Sigma_1 &= \sum_{s=s_{\min}}^{s_{\max}-1} \left(s \prod_{j=s_{\min}}^{s} \frac{f_j}{\bar{\mathbb{Z}} n_k} \right) , \\
\Sigma_2 &= \sum_{s=s_{\min}}^{s_{\max}-1} \left((s - s_{\min} + 1) \prod_{j=s_{\min}}^{s} \frac{f_j}{\bar{\mathbb{Z}} n_k} \right) , \\
\Sigma_3 &= \sum_{s=s_{\min}}^{s_{\max}-1} \left(s(s - s_{\min} + 1) \prod_{j=s_{\min}}^{s} \frac{f_j}{\bar{\mathbb{Z}} n_k} \right) .
\end{aligned}$$

The found value of $\bar{\mathbb{Z}}$ is used to calculate the ionization fractions according to the Eqs. 82 and 81.

The electron number density due to the considered element is found as $n_e = \bar{\mathbb{Z}} n_k$ (total electron density is found as a sum of electron densities for all elements).

7 Analysis

The subroutine `analysis` (in `analysis.F90`) is responsible for tracking position of the photosphere and calculating the bolometric luminosity, the main output of the code. Observed bolometric luminosity (`lum_observed` in the code) is calculated as suggested in [20] and consists of two parts, the luminosity at the photoshere and the luminosity due to the absorption of γ-rays from ^{56}Ni above the photosphere

$$L_{\rm obs} = L_{\rm photo} + \int_{m_{\rm photo}}^{M} S_{\rm dep}(m) dm \, . \tag{85}$$

Here $m_{\rm photo}$ is the mass coordinate of the photosphere, M is the total mass of the star, $S_{\rm dep}$ is the energy per gram per second deposited by γ-rays. The location of the photosphere is defined by the optical depth $\tau = 2/3$, and $L_{\rm photo}$ is found from the Eq. 4 at this point.

Among the quantities, useful for analysis and comparison of the simulations with previously done analytical work, SNEC calculates the position of the luminosity shell, characteristic diffusion and expansion times, as well as the internal energy of the shells. The idea of treating the expanding envelope of a supernova as a series of successive shells has been widely used in the theoretical studies, in particular, see [21] and [22]. Position of the luminosity shell is defined by the criteria $\tau \approx c/v$, given by the equality of the characteristic diffusion timescale

$$t_{\rm diff} \approx \frac{\kappa}{c} \rho \Delta r^2 \tag{86}$$

and expansion timescale

$$t_{\rm exp} \approx \frac{\Delta r}{\Delta v} \tag{87}$$

of a shell, where Δr is the shell's width and Δv is the difference between the velocities of its boundaries.

Having the bolometric luminosity and radius of the photosphere R_{ph}, SNEC calculates the effective black body temperature as $(L/4\pi\sigma R_{\mathrm{ph}}^2)^{1/4}$, where σ is the Stefan-Boltzmann constant. Following arguments of [23], we choose the effective temperature to be

$$T_{\mathrm{eff}} = \max\left\{ \left(\frac{L}{4\pi\sigma R_{\mathrm{ph}}^2} \right)^{1/4}, 5000 \right\} \mathrm{K}\,. \tag{88}$$

With this temperature we find the absolute magnitudes of a considered model in 11 different observed wavelength bands, using a table of bolometric corrections `BolCorr.dat` (see [24]). Together with time and T_{eff} they are saved in the output file `magnitudes.dat`. Here we need to make two important remarks. First, we cannot apply the above approach to the entire light curve, because at some point the whole ejecta becomes optically thin and the luminosity is dominated by the contribution of ^{56}Ni. For that reason the code finds the moment of time, when the luminosity contribution due to ^{56}Ni above the photosphere amounts to more than 5% of the total luminosity (arbitrarily chosen criteria), and we cannot trust the black body approach anymore. Secondly, it is known from the literature that in the real SN the U and B-bands of the light curves cannot be adequately reproduced by the one-temperature code, because already starting from a few tens of days these bands are strongly influenced by the iron group line blanketing.

References

[1] J. Von Neumann and R. D. Richtmyer. *J. Appl. Phys.*, **21**, 232, 1950.

[2] A. Mezzacappa and S. W. Bruenn. *Astrophys. J.*, **405**, 669, 1993.

[3] D. Mihalas and B. Weibel-Mihalas. *Foundations of Radiation Hydrodynamics*. Dover Publications, Mineola, NY, USA, 1999.

[4] M. C. Bersten. *Comparing Hydrodynamic Models with Observations of Type II Plateau Supernovae*. PhD thesis, U. of Chile, Santiago de Chile, Chile, 2010. arXiv:1303.0639.

[5] C. D. Levermore and G. C. Pomraning. *Astrophys. J.*, **248**, 321–334, 1981.

[6] M. C. Bersten, O. Benvenuto, and M. Hamuy. *Astrophys. J.*, **729**, 61, 2011.

[7] E. O'Connor and C. D. Ott. *Class. Quantum Grav.*, **27**, 114103, 2010.

[8] E. O'Connor. *Submitted to Astrophys. J. Supp. Ser. , arXiv:1411.7058*, 2014.

[9] R. L. Bowers and J. R. Wilson. *Numerical modeling in applied physics and astrophysics*. Jones and Bartlett, Boston, MA, USA., 1991.

[10] C. A. Iglesias and F. J. Rogers. *Astrophys. J.*, **464**, 943, 1996.

[11] J. W. Ferguson, D. R. Alexander, F. Allard, T. Barman, J. G. Bodnarik, P. H. Hauschildt, A. Heffner-Wong, and A. Tamanai. *Astrophys. J.*, **623**, 585, 2005.

[12] J. W. Ferguson and A. Dotter. Changing Abundances, Changing Opacities. In L. Deng and K. L. Chan, editors, *Proceedings of IAU Symposium 252*, page 1, 2008.

[13] D. A. Swartz, P. G. Sutherland, and R. P. Harkness. *Astrophys. J.*, **446**, 766, 1995.

[14] J. R. Kamm. Evaluation of the Sedov-von Neumann-Taylor Blast Wave Solution. Technical report, Los Alamos National Laboratory, Los Alamos, NM 87545 USA, 2010.

[15] B. Paczynski. *Astrophys. J.*, **267**, 315, 1983.

[16] A. Weiss, W. Hillebrandt, H.-C. Thomas, and H. Ritter. *Cox and Giuli's Principles of Stellar Structure*. Cambridge Scientific Publishers, Cambridge, UK, 2004.

[17] M. R. Zaghloul, M. A. Bourham, and J. M. Doster. *J. Phys. D.*, **33**, 977, 2000.

[18] F. X. Timmes and D. Arnett. *Astrophys. J. Supp. Ser.* , **125**, 277, 1999.

[19] F. X. Timmes and F. D. Swesty. *Astrophys. J. Supp. Ser.* , **126**, 501, 2000.

[20] T. R. Young. *Astrophys. J.*, **617**, 1233–1250, 2004.

[21] E. Nakar and R. Sari. *Astrophys. J.*, **725**, 904, 2010.

[22] A. L. Piro, P. Chang, and N. N. Weinberg. *Astrophys. J.*, **708**, 598, 2010.

[23] D. A. Swartz, J. C. Wheeler, and R. P. Harkness. *Astrophys. J.*, **374**, 266, 1991.

[24] E. O. Ofek. MATLAB package for astronomy and astrophysics. Astrophysics Source Code Library, 2014. URL http://adsabs.harvard.edu/abs/2014ascl.soft07005O.